

# Praxisbericht: Downgrade Datenbank Enterprise Edition auf Standard Edition One

Andrew Lacy, OPITZ CONSULTING GmbH

Woher weiß man, ob seine Applikation die Enterprise Edition (EE) wirklich benötigt? Welche Einschränkungen macht Oracle beim Downgrade der EE auf die Standard Edition One (SE1)? Gibt es versteckte Kosten bei einem solchen Migrationsvorhaben? Der Artikel gibt die Antworten darauf.



Ursprünglich wollte der Autor nur mit einem Kunden ein Konzept für die Migration von Oracle-Datenbanken auf eine virtuelle Umgebung erstellen. Als sie bei dieser Gelegenheit allerdings feststellten, dass der Kunde unterlizenziert war, fiel der Entschluss, statt sofort weitere Enterprise-Edition-Lizenzen zu kaufen, die Standard Edition One einmal näher anzuschauen.

Die Vorteile eines Downgrades auf die SE1 liegen klar auf der Hand: Es lässt sich Geld zu sparen – viel Geld sogar. Im Laufe von fünf Jahren würde der Kunde durch dieses Downgrade 95 Prozent an Supportkosten sparen (siehe *Abbildung 1*) Offizielle Infos zu den Technologiepreisen von

Oracle stehen unter „<http://www.oracle.com/us/corporate/pricing/technology-price-list-070617.pdf>“.

## Versteckte Kosten

Gibt es versteckte Kosten bei einem solchen Downgrade? Ja, die gibt es. Zum Beispiel, wenn es notwendig wird, Server-Hardware auszutauschen. Bei der Standard Edition darf man maximal vier Sockets auf dem Server-Motherboard haben, bei der SE1 sogar nur zwei. In diesem Fall nutzte der Kunde die Gelegenheit, im Zuge der geplanten Migration auf virtuelle Maschinen eine neue Hardware mit zwei Sockets zu kaufen.

Im Vorfeld war es natürlich wichtig, die ganze Umgebung miteinzubeziehen und sicherzustellen, dass der Betrieb auch nach der Migration auf SE1 weiterhin gut funktioniert. Aus diesem Grund wurden die Lösungen für folgende Betriebsstätigkeiten einmal näher betrachtet:

- **Backup/Recovery**  
Mithilfe der „Block Change Tracking“-Datei merkt sich die Enterprise Edition, welche Blöcke sich geändert haben, sodass ein Incremental Backup viel schneller erfolgen kann. Auf der SE1 gibt es diese Datei nicht, sodass das Incremental Backup hier viel langsamer vonstattengehen würde. Die Backup/Recovery-Skripte mussten also geändert werden.
- **Monitoring**  
Der Kunde verfügte bereits über eine Nagios-ähnliche Monitoring-Lösung.

Der Autor empfahl dennoch den Oracle Enterprise Manager Cloud Control (EMCC) als zusätzliches Überwachungs- und Administrationstool für die Oracle-Datenbanken. Das Tool besticht mit seiner benutzerfreundlichen Oberfläche und verständlichen Erklärungen – und kostet keine Extra-Lizenz. Es passt insbesondere für Unternehmen, die das Monitoring nur von 9 bis 17 Uhr brauchen, bei denen also der DBA in dieser Zeit verfügbar ist und den Screen verfolgen kann.

- **Housekeeping-Skripte**  
Für E-Mail-Benachrichtigungen ist die Enterprise Edition plus Diagnostic Packs erforderlich. Daher wurde bei der SE1 auf die Benachrichtigung durch den EMCC verzichtet. Dennoch nimmt der EMCC eine intensivere Prüfung vor als die Standard-Plug-ins von Nagios. Möchte man allerdings außerhalb der Anwesenheit des DBAs Warnhinweise erhalten, benötigt man neben EMCC eine separate Lösung – deswegen wurden die Nagios-ähnliche Lösung und EMCC gemeinsam zum Einsatz gebracht. Erstere wird dabei weiterhin als Standard-Monitoring-System für alle Server im Unternehmen eingesetzt, also nicht nur für Oracle.
- **Performance-Tools**  
Der Kunde hatte vor der Migration die Enterprise Edition mit Diagnostic Pack und Tuning Pack im Einsatz. Der zuständige Administrator konnte also mithilfe der Berichte des Automatic Workload Repository (AWR) Tuning-Probleme lö-

# Oracle Server	20	20% Rabatt	Support 22% Listenpreis	Supportkosten pro Jahr	Ersparnis im ersten Jahr
# CPUs	2				
# Cores	8				
Core Factor	0,5				
EE CPU	34.200 €	0,2	7.524 €	1.203.840 €	1.167.091 €
SE1 CPU	4.176 €	0,2	919 €	36.749 €	
EE Licence	4.377.600 €				
SE1 Licence	133.632 €				

Abbildung 1: Beispielrechnung zur jährlichen Ersparnis von Supportgebühren bei 20 Servern (Kurs: 0,72 Euro = 1 Dollar)

```
[oracle@deffbslor03 licence]$ ./licence_software.sh
detect oracle version
List Oracle software options installed:
-----
Context Management Text = ON
Oracle Label Security = OFF
Oracle Database Vault = OFF
Oracle OLAP = OFF
Oracle Partitioning = OFF
Oracle Data Mining = OFF
Oracle Real Application Testing = OFF
Oracle Real Application Cluster = OFF
Oracle ASM = OFF
[oracle@deffbslor03 licence]$
```

Listing 1

```
Option/Management Pack |Used |DB Version
|ReportGen Time
-----|-----|-----|
Active Data Guard |NO |12.1.0.1.0
|2014-05-07 14:11:14
Advanced Compression |NO |12.1.0.1.0
|2014-05-07 14:11:14
Advanced Security |NO |12.1.0.1.0
|2014-05-07 14:11:14
Change Management Pack |NO |12.1.0.1.0
|2014-05-07 14:11:14
Configuration Management Pack for Oracle Database |NO |12.1.0.1.0
|2014-05-07 14:11:14
Data Masking Pack |NO |12.1.0.1.0
|2014-05-07 14:11:14
```

Listing 2

sen. Mit der SE1 muss er auf AWR verzichten, stattdessen steht Statspack zur Verfügung, das eingeschränktere Möglichkeiten bietet. Weitere Performanzprobleme waren nach der Migration der EE auf SE1 zu erwarten, weil es dort keine Parallel-Funktionalität gibt. Das heißt, jeder Prozess nimmt ei-

nen Core, wenn er frei ist. Aber in der Enterprise Edition kann durch Parallel Processing ein Prozess über mehrere Cores geteilt werden. Das bringt unter Umständen einen großen Vorteil in der Performance. Es musste also eine zusätzliche Lösung her, um die zu erwartenden Performanzprobleme besser in

den Griff zu bekommen: Sie wurde mit der TOAD DBA Suite von Dell Software (ehemals Quest) gefunden, die auch für SE1 funktioniert.

Die Migration von zwanzig Datenbanken hat bei diesem Projekt acht Prozentange in Anspruch genommen, die sich über fünf Tage erstreckten. Der Kunde hatte leider die SE1-Lizenzen gekauft und die EE-Lizenz gekündigt, bevor die Datenbank-Migration geplant war. Als dann ein Brief von Oracle ankam, der ein Lizenz-Audit ankündigte, musste man auf die Schnelle alle zwanzig Datenbanken migrieren.

### Braucht die Applikation wirklich EE?

Wie kann man wissen, ob eine Applikation die Enterprise Edition wirklich braucht? Die erste Frage, die sich hier stellt, lautet, ob eine der besonderen Optionen der EE genutzt wird. Die Enterprise Edition wird mittlerweile mit 24 extra kostenpflichtigen Optionen angeboten (siehe „[www.oracle.com/us/products/database/enterprise-edition/comparisons/index.html](http://www.oracle.com/us/products/database/enterprise-edition/comparisons/index.html)“ und „<http://www.oracle.com/us/products/database/options/overview/index.html>“). Einige der bekanntesten sind:

- HA, OLAP, RAT
- Partitioning
- Komprimierung
- Sicherheit
- Spatial
- Multitenant
- In-Memory

Die nächste Frage ist, ob die Zusatz-Optionen ein- oder ausgeschaltet sind. Die MOS-Notes 1069015.1, 942406.1, 948061.1 und 942406.1 erklären, wie man einen Datenbank-Server so installiert, dass es nicht möglich ist, verschiedene Optionen zu benutzen – eine sehr nützliche Einstellung, wenn diese nicht lizenziert sind. Dort ist außerdem erklärt, wie man herausfindet, ob die Optionen in der Software noch eingeschaltet sind. Optionen, die in der Software ausgeschaltet sind, können in der Datenbank nicht benutzt werden. Listing 1 zeigt den Output eines Skripts, das auf den angegebenen MOS-Notes basiert.

Ein weiterer wichtiger Punkt ist herauszufinden, welche allgemeinen Features aktuell genutzt werden. In diesem Fall be-

trieb der Kunde teilweise selbst geschriebene Applikationen. Im direkten Gespräch mit den Entwicklern kam heraus, welche Features in Benutzung waren. Auch die Hersteller von Applikationen können hierüber Auskunft geben. Sollten beide Informationsquellen nicht ausreichen, bleibt die Möglichkeit, diese Infos mit Skripten aus der Datenbank herauszulesen.

Entwickler und Hersteller gaben grünes Licht für die SE1. Sie selbst beziehungsweise der DBA stellten eine Liste von EE-Features zusammen, die in den Applikationen benutzt wurden:

- Parallel Query
- Transportable Tablespaces (TTS)
- Materialized View (MV)
- Virtual Private Database (VPD)
- Table Compression
- Bitmap Index

Das größte Problem für die Entwickler war die Virtual Private Database. Immerhin hing die Sicherheit einer Applikation davon ab. Diese Änderung nahm viel Zeit in Anspruch. Letztendlich stellte sich heraus, dass es immer ein paar Applikationen gibt, von denen niemand weiß, ob sie EE-Features benötigen oder nicht. Mit Skripten von Oracle konnten wir diese Information teilweise aus der Datenbank herausle-

```
LIST_OF_USERS=('ANONYMOUS','APEX_030200','APEX_040200','APEX_LISTENER',
,'APPOSSYS','CTXSYS','DBSNMP','DIP','DMSYS','EXFSYS','FLOWS_FILES','IA
CYEXT','MDDATA','MDSYS','MIGRATE_USER','MGMT_VIEW','PERFSTAT','OLAPSYS',
,'ORACLE_OCM','ORDDATA','ORDPLUGINS','ORDSYS','ORDDATA','OUTLN','OWBSYS',
,'OWBSYS_AUDIT','PUBLIC','SCOTT','SI_INFORMTN_SCHEMA','STDBY','SYS','S
YSMAN','SYSTEM','TMSYS','WMSYS','XDB','XS$NULL')
```

Listing 3

```
select listagg(username, ',') within group (order by username)
from dba_users@migrate_dblink
where username not like 'OPSS%'
and username not in ${LIST_OF_USERS}
and username in (select distinct owner from dba_objects@migrate_dblink)
and account_status = 'OPEN'
order by 1;
```

Listing 4

sen. Listing 2 zeigt ein Beispiel-Output des Skripts in MOS Note 1309070.1; leider liefert es nur begrenzt Informationen über die EE-Benutzung, weil nicht jedes EE-Feature geprüft wird.

### Die Migration

Wer sich entschieden hat, auf die Standard Edition One zu migrieren, sollte auch das richtige Vorgehen für das Downgrade kennen. Es gibt nämlich nur

eine Methode, die Oracle unterstützt: die Migration via Export//Import. Im Detail akzeptiert der Hersteller die folgenden Vorgehensweisen:

- Export durch „exp“, Import durch „imp“
- Export durch „expdp“, Import durch „impdp“
- Wenn kein Long-Spalten-Typ vorhanden ist, geht auch „impdp“ mit „dblink“ (ohne „expdp“)

```
select 'create user ' || name || ' identified by values ' || chr(39) || spare4 || ';' || password || chr(39)
||';'
from sys.user\${migrate_dblink}
where type# = 1
and password is not null and spare4 is not null
and name not like 'OPSS%'
and name not in ${LIST_OF_USERS}
union
select 'create user ' || name || ' identified by values ' || chr(39) || password || chr(39) ||';'
from sys.user\${migrate_dblink}
where type# = 1
and password is not null and spare4 is null
and name not like 'OPSS%'
and name not in ${LIST_OF_USERS}
union
select 'create user ' || name || ' identified by values ' || chr(39) || spare4 || chr(39) ||';'
from sys.user\${migrate_dblink}
where type# = 1
and password is null and spare4 is not null
and name not like 'OPSS%'
and name not in ${LIST_OF_USERS}
order by 1;
```

Listing 5

```

if my_rownum > 1 then
  db_string:='alter tablespace ' || chr(34) ||
    my_tablespace_name || chr(34) || ' add datafile ' ||
    chr(39) || replace(replace(replace(my_file_name,
    '${OLD_DIRS1}', '${NEW_DIRS1}'), '${OLD_DIRS2}',
    '${NEW_DIRS2}'), '${OLD_DIRS3}', '${NEW_DIRS3}') ||
    chr(39) || ' size ' || my_bytes || ' reuse;';
  dbms_output.put_line(db_string);
else
  select count(*) into my_count from dba_data_files
  where tablespace_name=my_tablespace_name;
  if my_count > 0 then
    db_string:='commit;';
    dbms_output.put_line(db_string);
    db_string:='drop tablespace ' || chr(34) ||
      my_tablespace_name || chr(34) || ' including contents and
      datafiles cascade constraints;';
    dbms_output.put_line(db_string);
  end if;
  db_string:='create tablespace ' || chr(34) ||
    my_tablespace_name || chr(34) || ' datafile ' || chr(39) ||
    replace(replace(replace(my_file_name, '${OLD_DIRS1}',
    '${NEW_DIRS1}'), '${OLD_DIRS2}', '${NEW_DIRS2}'),
    '${OLD_DIRS3}', '${NEW_DIRS3}') || chr(39) || ' size ' ||
    my_bytes || ' reuse;';
  dbms_output.put_line(db_string);
end if;

```

Listing 6

```

# added VERSION=LATEST to the impdp to avoid error ORA-39021
# necessary when compatible on source is e.g. 11.2.0.3.5 (with last PSU number filled in)
impdp '/' as sysdba" DIRECTORY=${THE_DATAPUMP_DIR} logfile=importdp_${ORACLE_SID}.log NETWORK_LINK=migrate_
dblink SCHEMAS=${TESTIT} VERSION=LATEST | tee -a $LOG_OUTPUT

```

Listing 7

```

# added VERSION=LATEST to the impdp to avoid error ORA-39021
# necessary when compatible on source is e.g. 11.2.0.3.5 (with last PSU number filled in)
impdp '/' as sysdba" DIRECTORY=${THE_DATAPUMP_DIR} dumpfile=${DUMPY_FILE} logfile=LOG_DIR:importdp_${ORACLE_
SID}.log job_name=Import_${ORACLE_SID}_xx parallel=1 SCHEMAS=${TESTIT} | tee -a $LOG_OUTPUT

```

Listing 8

In diesem Fall entschied man sich für „impdp“ mit „dblink“ und dort, wo der Long-Spalten-Typ vorhanden war, für „expdp“/„impdp“.

### Auf dem alten Server

Auf dem alten Server ist ein Datenbank-Nutzer mit DBA-Rechten für den Database-Link erforderlich. Dieser wird angelegt;

dann wird geschaut, ob noch Database-Links zu anderen Datenbanken vorhanden sind – diese würde man dann auch für die neue Umgebung anlegen.

Wegen des knappen Zeitplans blieb man bei dem vorhandenen Character Set. Im Idealfall hätte man die Chance genutzt und dieses auf „AL32UTF8“ migriert. Im vorliegenden Fall musste man herausfin-

den, welches Character Set derzeit in Gebrauch war, und die neue Datenbank damit anlegen. Wo Long-Spalten vorhanden waren, erfolgte gleich der Export durch „expdp“.

### Auf dem neuen Server

Die neuen Server sind virtuelle Maschinen. Es lagen VM-Templates für die Da-

tenbank-Versionen 10.2 und 11.2 vor. Da man in dem Projekt in der glücklichen Lage war, den alten und den neuen Server parallel vorliegen zu haben, war eine Testmigration für jede Datenbank möglich. Die Migration ließ sich durch Skripte für alle nötigen Schritte automatisieren. Das sparte sehr viel Zeit und garantierte, dass die Migration reibungslos ablief.

Bei einer Testmigration kann es vorkommen, dass Daten nicht vollständig sind. Wir importierten ja von einer Produktiv-Datenbank, auf der Änderungen stattfinden. Um dieses Problem bei der Produktionsmigration zu umgehen, wurden der Listener Port geändert und die Datenbank-Instanz durchgestartet.

Im weiteren Verlauf legte man zunächst einen Datenbank-Link mit dem Benutzer an, der auf dem alten Server eingerichtet war. Diesen Link prüfte man in dem Zuge auch gleich. Jetzt konnte man Informationen vom alten Server auslesen, um vor allem herauszufinden, welche Schemata migriert werden mussten. Danach wurden die Datenbank-Objekte Roles, Grants, Tablespaces, Profiles, Directories, User, OPS User, Public Synonyms und User Grants erstellt.

- Roles, Grants
- Tablespaces
- Profiles
- Directories
- User
- OPS User
- Public Synonyms
- User Grants

Im Anschluss konnte man den Import über den Datenbank-Link laufen lassen und anschließend ein „Recompile“ aller „Invalid“-Objekte durchführen. Am Ende wurde der Import gecheckt, indem man die Anzahl der Objekte auf beiden Servern verglich.

### Schemata und Benutzer

Der schwierigste Teil der Migration war das Skript, bei dem es um die Bestimmung der Schemata beziehungsweise User geht. Der Autor fing mit einer Variablen an, die alle Nutzer enthält, die er definitiv nicht importieren wollte. Dabei handelt es sich um Nutzer, die standardmäßig angelegt sind, oder um Benutzer, die nur vorhanden sind, wenn besondere

Optionen wie „Spatial“ oder „OLAP“ installiert werden.

Die Funktion „listagg“ lieferte eine Komma-separierte Liste der Benutzer, die zu migrieren waren (siehe Listing 3). „OPS\$“-Benutzer legte er separat an. Die zu migrierenden Benutzer müssen eigene Objekte haben, ansonsten werden diese automatisch durch „impdp“ erstellt. Sie waren also nicht gesondert zu importieren (siehe Listing 4).

Da es in der Version 10.2 keine „listagg“-Funktion gibt, wurde Ersatz dafür gefunden (siehe <http://www.oracle-base.com/articles/misc/string-aggregation-techniques.php>).

### User-Passwörter und Tablespaces

Auf einer 11.x-Datenbank existieren drei verschiedene Arten von Passwörtern. Ein Skript lieferte die richtigen (siehe Listing 5). Jedes Data File, das im alten Server vorhanden war, wurde auch im neuen angelegt. Dabei erstellte man mit einem einfachen Trick auch die Tablespaces: Wenn es sich um das erste Data File des Tablespace handelte, erstellte man den Tablespace, andernfalls wurde dieses Data File zu den Tablespaces hinzugefügt (siehe Listing 6).

Im Anschluss an die genannten Schritte startete der Import über den Datenbank-Link (siehe Listing 7). Listing 8 zeigt das Import-Kommando, sofern Long-Spalten vorhanden sind.

### Weitere Änderungen

Nachdem die Datenbanken nach Standard Edition One migriert waren, fielen noch einige weitere Änderungen an, um einen normalen Betrieb zu garantieren. Eine betraf das Backup-Skript. Es musste modifiziert werden, weil es bislang Incremental Backups benutzt hatte. Wie eingangs erwähnt, sind Incremental Backups auf der EE eine schnellere Lösung. Auf der SE1 hingegen führen die Incremental Backups immer einen Full Database Scan durch.

Fürs Monitoring wechselte man auf den Enterprise Manager Cloud Control von Oracle, der per Default so viele Regeln liefert, dass er einen besseren Monitoring-Job macht als die meisten Nagios-Plug-ins. Unter SE1 musste lediglich auf die Benachrichtigungsfunktion ver-

zichtet werden, weil dafür die Enterprise Edition plus Diagnostic Pack erforderlich wäre. Da sich die Performance teilweise viel schlechter darstellte und der Administrator daran gewöhnt war, mit AWR-Berichten die Probleme zu lösen, wurde ihm eine andere Lösung zur Verfügung gestellt.

### Fazit

Als Folge dieses Projekts konnte der Kunde viel Geld sparen. Nicht ganz so erfreulich waren die Entwicklungskosten, die für eine Alternativlösung zur Virtual Private Database zu hoch ausfielen. So war es zeitweise unumgänglich, gleichzeitig Supportkosten für das alte und das neue System zu zahlen. Einen weiteren Knackpunkt bildete das Monitoring, das mit EMCC nun eingeschränkter funktioniert, weil keine E-Mail-Benachrichtigungen möglich sind. Auch mit einer teilweise schlechteren Performance musste der Kunde leben.

Die Entwicklungskosten stellten aufgrund des insgesamt eingesparten Geldes kein Problem dar. Aufgrund des Nagios-Monitorings brachte auch die eingeschränkte EMCC-Funktionalität keine Nachteile. Die Performance-Probleme waren schon vor dem Downgrade erwartet worden.

Durch die große Ersparung wurden das Projektziel gut erreicht und die Erwartungen des Kunden erfüllt. Hinzu kam, dass Oracle auf die Lizenz-Überprüfung verzichtete, weil das Unternehmen des Autors als Oracle Platinum Partner die Migration durchgeführt hatte.



Andrew Lacy  
andrew.lacy@opitz-consulting.com